מבוא

חדשנות בהצגת נושאי המחשב ועבוד נתונים

עם ההתפתחות המהירה של יישום נושאי עבוד נתונים במחשב, בפרט במעבר מהשטח המדעי לשטח המנהלי/מסחרי, הורגש הצורך להסביר את יסודות המחשב ועבוד נתונים לאוכלוסיה רחבה של אנשים בגילים שונים וברווריציה של השכלה , התמחות ותפקוד.

עד כה התבססה ההדרכה שניתנה לאוכלוסיה זו על הגישה הקלסית : מתחילים מהיסודות המקובלים (מספרים בינריים, מבנה המחשב וכו') ועולים הלאה. בדרך יסורים זו, לא רק שלא מגיעים לעיקר (שהוא יסודות עבוד נתונים) אלא משתעממים ומתיאשים מהנושא. שיטת ההדרכה הקלסית במחשב דומה לשיטת הקריאה הקלסית שהונהגה עד לפני דור אחד : מתחילים בלמידת האותיות, עוברים להברות ולבסוף מגיעים בדרך ארוכה לקריאת המלים.

השיטה החדשנית בהדרכת נושא המחשב מקבילה לשיטה החדישה בלמוד הקריאה : דהיינו מתחילים לקרוא מלים לפני שלומדים אותיות והברות. בשיטה זו אפשר להציג את נושא המחשבים ועבוד נתונים בצורה רחבה ויעילה תוך חמש-שש שעות :

מתחילים בתאור האמצעים לאחסון האינפורמציה , עוברים לרשומה ולגוש, מסבירים את שיטות האחסון והשליפה של הרשומות והספריות , ובסוף עוברים למבנה המחשב ודרך פעולתו עד לעבודה בשפה עילית אחת (כמו פורטרן) .

הפרקים הבאים נכתבו בצורה שישמשו הן להכוונת המרצה והן כחומר-עזר מתומצת שיחולק לחניכים בתום כל הרצאה.

בנוסף פרקים נוספים ניתנים כנספח ומיועדים לעיון עצמי.
לקראי מבוא כללי על השפות PL1 , COBOL , BASIC

ובהן מאיר.

הערה: השימוש בכרטיסי I.B.M. בקלט ובפלט נפסק כמעט בזמן האחרון.
הצגים החליפו את הכרטיסים (בערך כל שורה של 80 תווים לכרטיס).
כמו כן צומצם מאוד השימוש בנייר להדפסת הפלט. רוב הפלט
מאוחסן על דיסקים ומוצג ע"י הצג.

1. אמצעי אחסון והעברת אינפורמציה

א. <u>הגדרות</u> :

מלה מורכבת מאותיות. מספר מורכב מספרות (או מספרה אחת). סדרה של אותיות היא סדרה אלפבתית. סדרה של ספרות היא סדרה נומרית. סדרה מעורבת של אותיות וספרות היא סדרה אלפנומרית. בהדפסה מופיעים סימנים מיוחדים בנוסף לאותיות והספרות : למשל סימני הפסוק והסימן רווח. האותיות , הספרות והסימנים המיוחדים , כולם ביחד, נקראים תווים (CHARACTERS). השם שניתן , במחשבי י.ב.מ. לתוו הוא ה- "בית" (BYTE).

ב. <u>כרטיס י.ב.מ.</u>

כרטיס י.ב.מ. (בגודל 18.75x8.25 ס"מ) מכיל 12 שורות ו - 80 טורים. הכרטיס סימטרי גם מצדדים (ימין - שמאל) וגם מעלה - מטה. הטורים ממוספרים משמאל לימין מ-1 עד 80.

השורות נקראות מלמעלה למטה : 12, 11, 0, 1, 2 ....עד 9 .

טור אחד בכרטיס מיועד לתת ניקוב לתוו אחד. המקביל לתוו רווח הוא טור ללא ניקוב. אם התוו הוא ספרה אזי הניקוב הוא של אותה ספרה בלבד.

שורות הכרטיס מחולקות לשני חלקים : שלש השורות העליונות (דהיינו 12, 11, 0) נקראות "אזורים" (ZONES). תשע השורות התחתונות (דהיינו 1, 2,....עד 9) נקראות ספרות (FIGURES). בד"כ מנוקבת האות(בלטינית ובעברית) בנקוב אחד באזורים ובנקוב אחד בספרות.

נקוב של 12 לבד (באזורים) משמעותו הסימן "&" בלטינית והאות "א" בעברית.

נקוב של 11 לבד (באזורים) משמעותו הסימן "-", דהיינו "מינוס".

כמובן ניקוב של "0" לבד משמעותו הספרה אפס.

להלן נקוב האותיות והתוו "/" (קו נטוי) בלטינית ובעברית :

12	12	12	12	12	12	12	12	12	11	11	11	11	11	11	11	11	11	0	0	0	0	0	0	0	0	0
1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	/	S	T	U	V	W	X	Y	Z
ב	ג	ד	ה	ו	ז	ח	ט	י	ך	כ	ל	ם	מ	ן	נ	ס	ע	/	ף	פ	ץ	צ	ק	ר	ש	ת

<u>הערה</u> : מחשבי י.ב.מ. מכירים ניקובים חוקיים של 256 תווים. רק חלק מהתווים ניתנים להדפסה. כמו כן רק חלק מהתווים ניתנים לניקוב במקש אחד במכונת ניקוב. לחלק מהתווים החוקיים (שהם לא אותיות ולא ספרות) יותר משני ניקובים בטור אחד.

ג. <u>סרט נייר מחורר</u>

קיימים סרטים ברחבים שונים עבור 5, 6, 7, 8 ערוצים (של אינפורמציה).
בכל הרחבים מנוקבים התווים בצפיפות של 10 תווים לאינטש.
הערוצים נקראים מלמטה למעלה : 1, 2, 4, 8, 16, 32, 64, 128.
סרט של 5 – ערוצים מכיל את חמשת הערוצים 1 עד 16. בכל הרחבים קיים גם ערוץ נוסף של מורה דרך, מעל שלשת הערוצים התחתונים , דהיינו בין ערוץ 4 לערוץ 8.

בסרט של 8 ערוצים (רוחב 1 אינטש) מנקבים תו ע"י ניקוב או אי-ניקוב של עגולים (בגודל זהה) בכל אחד משמונת הערוצים. ניקובים של עגולים יותר קטנים תמיד מופיעים בערוץ מורה הדרך, כדי להצביע על מקום המצאות התווים – ניקובים אלו מורחשים ע"י גלגל השיניים בקוראי-הסרטים האלקטרו-מגנטיים או ע"י האור העובר דרכם בקוראי-הסרטים הפוטו-אלקטריים.

בסך הכל ישנם 9 ערוצים בסרט הנייר הרחב והם סימטריים במקומם מעלה-מטה. כל טור המכיל ניקוב של מורה-דרך נקרא FRAME והוא מיועד לתו אחד. כל אחד מ- 8 הניקובים (או אי-הניקובים) המרכיבים את התו נקרא סבית (BIT). ישנן 2^8 דהיינו 256 אפשרויות לנקוב התו. אלו מקבילים ל- 256 התווים החוקיים במחשבי י.ב.מ. (הניתן לתרגם לניקוב טורים בכרטיסים).
הניקוב בסבית נקרא "1" . האי-ניקוב בסבית נקרא "0" .

ד. <u>סרט מגנטי</u>

זהו סרט הדומה לסרט קסטה , רק ברוחב חצי אינטש. הכתיבה עליו דומה לזו שעל סרט נייר מחורר של 8 ערוצים : במקום לחורר את הסביות "1", "0" נותבים סימנים מגנטיים. בנוסף ל- 8 ערוצי האינפורמציה ישנו גם ערוץ של אימות שנקרא ערוץ הזוגיות(PARITY).
אם עובדים בשיטת ה- EVEN-PARITY דואגים שסך הכל האחדים בתור (BYTE/FRAME)-כולל זה שבערוץ ה- PARITY – יהיה זוגי.
ואם עובדים בשיטת ה- ODD-PARITY, הסך הכל יהיה בלתי-זוגי.
סרט מגנטי כזה נקרא 9-TRACK , דהיינו ערוץ ה- PARITY נוסף.
כותבים על הסרט המגנטי בצפיפויות שונות. מקובלות כיום צפיפויות של 800, 1600, 6250 תווים לאינטש.
כונני הסרטים המגנטיים קוראים וכותבים על הסרטים במהירויות שונות – מעל 100,000 תווים לשניה.

ה. <u>התקליטון המגנטי (דיסקט).</u>

על תקליטון קטן (שקוטרו כרוחב דף זה) אפשר לכתוב ולקרוא אינפורמ-ציה (בצורה מגנטית). קבול התקליטון כרבע-מיליון תווים (כ-3000 כרטיסים). הכתיבה והקריאה עליו נעשית דרך התקלידית - מכובה עם מקשים וצג של שורה אחת (דהיינו [illegible] כרטיס אחד).

ו. <u>התקליטים המגנטיים (דיסקים)</u>

נתאר דיסק נפוץ בקבול של 100 מיליון תווים. דיסק זה עשוי שכבות (מופרדות) של תקליטים שכותבים עליהם מגנטית משני הצדדים בעגולים קונצנטריים שנקראים מסלולים (TRACKS). ישנם 20 ראשים קוראים/כותבים הנעים בזרועות כדי להגיע לכל אחד מהעיגולים (שמספרם 400 ועוד 7 רזרבה). משטח אחד מתוך 20 המשטחים מכיל אינפורמציה על מס' המסלול וכו'. שאר 19 המסלולים הנמצאים זה מעל זה מהווים יחידה הנקראת גליל (CYLINDER).

בגלל תנועת הזרועות רצוי להתקדם באחסון האינפורמציה מגליל לגליל. בדיסק המתואר קבול הגליל הנו כרבע-מיליון תווים וקבול המסלול הוא כ- 13000 תווים.

ז. <u>הצגים DISPLAYS</u>

הצגים הנפוצים מכילים מסך טלוויזיוני המציג 24 שורות של 80 תווים. על יד המסך נמצא לוח מקשים. הצג משמש להעברת אינפורמציה מאת ואל המחשב. (מסך רגיל: Cathode Ray Tube; מסך חדש Liquid Crystal Display

ח. <u>יחידות קלט-פלט שונות</u>

מכשיר המתחבר למחשב ומעביר אינפורמציה אליו נקרא יחידת-קלט. לעומת זאת מכשיר המתחבר למחשב ומקבל ממנו אינפורמציה נקרא יחידת-פלט.

ישנן יחידות המשמשות לשני התפקידים יחד, כגון כונן התקליטון, כונן הדיסק, כונן הסרט המגנטי.

וישנן יחידות המשמשות לתפקיד אחד בלבד, כגון קורא-כרטיסים, מנקב-כרטיסים, קורא סרט נייר מחורר, מנקב סרט נייר מחורר, מדפסת תווים, מדפסת שורות, קורא אופטי (של טקסט מודפס) וכותב על מיקרופילם.

יחידות הקלט-פלט משתמשות באמצעי האחסון של אינפורמציה. ישנם אמצעים העבירים מצורה לצורה: למשל אינפורמציה של כרטיס אפשר להעביר לסרט מגנטי או לדיסק וכו' (דרך מחשב).

ישנם אמצעים בלתי-עבירים מצורה לצורה: למשל אינפורמציה המודפסת על דף נייר, אי-אפשר להעבירה, דרך מחשב, לכרטיסים אלא יש לקרוא אותה ולנקב אותה מחדש.

ט. <u>אינפורמציה אנלוגית ואינפורמציה ספרתית:</u>

תמונות, קולות ומוסיקה הן אינפורמציה אנלוגית שאפשר להמיר לאינפורמציה ספרתית (אפסים ואחדים בינריים) ולאחסן אותה כך או להעבירה למקום רחוק. ההפך נעשה ע"י digitizers אינפורמציה ספרתית אפשר להמיר לאינפורמציה אנלוגית ע"י מכשיר נגדי.

[illegible]

2. עבוד נתונים

א. הכרטיסיה ומונחי עבוד נתונים

בכנסים לשלם עבור גז והפקידה מחפשת את כרטיס הצרכן מתוך אוסף של כרטיסים שיש לה (לצרכניה באותה עיר). האוסף נקרא "כרטיסיה".
הכרטיס מכיל אינפורמציה בשטחים שונים : שם הצרכן, כתובתו, מס' הצרכן וכו' .
דוגמא זו ממחישה את יסוד עבוד נתונים. נשאל מהמונחים של הכרטיסיה כדי להגדיר מונחים טכניים.
כל הכרטיסיה נקראת "קובץ" (FILE) או "קובץ נתונים"(DATA SET).
כל כרטיס, המהווה יחידת אינפורמציה, נקרא רשומה (RECORD). אם הכרטיסיה היא גדולה (כעשרים אלף כרטיסים), לא ניתן לאחסן אותה (את כל הכרטיסים) בקופסה אחת. מאחסנים אותה בכמה קופסאות (עדיף בקבול שוה - נניח 1000 כרטיסים בקופסה). הקופסה משמשת כאן כאמצעי אריזה/אחסנה של הכרטיסים. קופסה כזו נקראת בעבוד נתונים "גוש" (BLOCK).

ב. הרשומה והשדות

אמצעי מוכר לרשומה הוא הכרטיס. רשומה יכולה לתפוס כרטיס שלם, חלק מכרטיס, או כמה כרטיסים ביחד.
הרשומה יכולה להיות באורך קבוע או באורך משתנה. הנתונים בתוך הרשומה מחולקים לשדות : למשל שדה אחד אלפבתי עבור השם ושדה שני נומרי עבור המספר. אורך הרשומה הוא מספר התווים המהווים אותה. אם הרשומה היא באורך משתנה, מוסיפים לה (בהתחלה) 4 תווים המכילים את האורך שלה (החדש, כולל 4 התווים).
הרשומות יכולות להתאחסן בכל אמצעי האחסון הקשורים למחשב : כרטיסים, סרט נייר מחורר, סרט מגנטי, דיסק וכו' .
הערה : בזמן האחרון מרבים להשתמש במונח "מסמך" ~~[illegible]~~ "רשומה" מסוג נפוץ.

ג. הגוש

מושג הגוש כאמצעי אריזה מקבל משמעות נוספת כאשר הרשומות נמצאות על סרט מגנטי או דיסק.

אם הרשומות הם באורך משתנה, גם הגוש יהיה באורך משתנה - כאן מוסיפים 4 תווים בהתחלת הגוש שיכילו את האורך החדש של הגוש, כולל 4 התווים.
הגוש על סרט מגנטי מהווה מנה אחת של קריאה או כתיבה. ישנו מרווח ניכר בין גוש לגוש. נניח שהצפיפות היא 1600 תווים לאינטש והמרווח הבין גושי הוא 1 אינטש, אזי אפשר לכתוב 20 רשומות של 80 תווים על 1 אינטש והסרט ינוצל מעשית רק במחציתו.

אם הגושים יהיו גדולים, ניצול הסרט יהיה גדול יותר אך לעומת זאת מבנה הקריאה/כתיבה מגדל - כדי לקרוא רשומה מסויימת נהיה מוכרחים לקרוא את כל הגוש הגדול המכיל אותה. לכן אורך הגוש נקבע כפשרה בין ניצול מקסימלי לשטח באמצעי האחסון לבין המהירות והיעילות בשליפת הרשומות.

ד. הקובץ

כפי שהוסבר לעיל הקובץ הוא אוסף כל הרשומות (הארוזות בגושים).

ה. שיטות האחסון ושליפת רשומות

1) האחסון הסקוונציאלי (SEQUENTIAL)

בשיטה זו הרשומות מתאחסנות לפי סדר שליחתם על אמצעי הפלט ("ארוזות" בגושים). שיטה זו טובה לכל אמצעי האחסון.

2) האחסון הישיר (DIRECT).

שיטה זו מהירה לשליפה אך מוגבלת לשמוש בדיסק כאמצעי אחסון. היא טובה כאשר ישנו מספר קבוע של רשומות באורך קבוע. לדוגמא נקח קובץ הלקוחות של חברה המכילה 2000 רשומות בנות 160 תווים כ"א. לכל לקוח יש מספר סדורי. נדרשים יום-יום שינויים ברשומות של כמה לקוחות. בשיטת האחסון הישיר מקצים בהתחלה מקום לכל הרשומות. אם רוצים לתקן רשומה עבור לקוח, שולפים את הרשומה (יודעים את מקומה כפונקציה של מספרה הסדורי), מתקנים אותה ומחזירים אותה לאותו מקום. כמובן את הרשומה הראשונה עבור הלקוח כותבים אותה ישר במקום המיועד לה.

3) האחסון בשיטת האינדקס-סקוונציאלי (INDEX-SEQUENTIAL).

שוב שיטה זו מוגבלת לשמוש בדיסק כאמצעי אחסון. הדוגמא הבאה תמחיש את השיטה:

ערכנו ספר עובדים במפעל גדול, ממויין לפי שם המשפחה והפרטי של העובד. הספר כתוב בשלש חוברות. בסוף כל חוברת יש מקום להוספות. כל חוברת מחולקת לפרקים. בסוף כל פרק יש מקום לתוספות. בכל דף הושאר מקום רזרבי לתוספות. אם רוצים להוסיף שם נוסף, מחפשים את מקומו לפי סדר המיון ושם תוקעים סימן "להסתכל בסוף הדף". בסוף הדף רושמים את השם הנוסף ומסמנים לאן ללכת בחזרה להמשך סדר המיון. כאשר המקום הרזרבי בסוף הדף מתמלא, הולכים למקום הרזרבי בסוף הפרק (עם הסימונים לאן ללכת ולאן לחזור). כאשר המקום הרזרבי בסוף הפרק מתמלא, הולכים למקום הרזרבי בסוף החוברת.

בדוגמא הנ"ל מסמל הדף "מסלול" על הדיסק. הפרק מסמל "גליל" והחוברת מסמלת דיסק שלם.

הערה: כאשר הקובץ מכיל יותר מדי תיקונים, מעתיקים אותו נקי במקום חדש (פעולה זו נקראת: CONDENSE).

4) <u>שיטת הספריה</u> ((LIBRARY=PARTITIONED DATA SET

גם שיטה זו מוגבלת לשמוש הדיסק כאמצעי אחסון, נבין את השיטה בדוגמא הבאה : יש לנו מגילת נייר ארוכה ממוספרת לפי שורות (נניח מ- 1 עד 50000). מצפים לכתוב הגדרה של כ- 2000 מרבזים במספר משתנה של שורות לכל מרבז .

נקצה את 2001 השורות הראשונות לשטח הכוונה (או מילון בשם DIRECTORY). שאר השטח יוקצה להגדרות ונקרא שטח הטקסט TEXT). בשורה הראשונה נכתוב את מספר השורות החופשיות (בהתחלה מ-2002 עד 50000). נתצר לצר שם ראשון שמוגדר בארבע שורות. נרשום את השם בשורה 2 עם הציון שהוא מוגדר בארבע שורות החל משורה 2002, ושם אכן נגדיר אותו. את השורה הראשונה נתקן כי המקום החופשי עתה הוא מ- 2006 עד 50000. את ההגדרה הבאה נרשום החל משורה 2006 (והשם בשורה 3 - כמובן עם תיקון שורה 1 בהתאם), וכך הלאה. אם רוצים לבטל הגדרה, אין צורך למחוק אותה מהטקסט, מספיק למחוק את שורת השם. אם רוצים לשנות הגדרה, קודם מבטלים אותה ואחר-כך כותבים אותה מחדש (בסוף המילון ובסוף הטקסט). במשך הזמן כאשר יתרבו הבטולים יהיה כדאי להעתיק את כל הספריה בצורה נקיה לשטח חדש. פעולה זו נקראת CONDENSE.

הערה : אפשר לקרוא בצורה סקוונציאלית רשומות שנערכו בכל אחת מארבע השיטות הנ"ל.

3. <u>המחשב</u>

א. <u>הגדרת מונחים</u>

יושב אדם ליד שולחן (יחידת הבקרה) שעליו מונחים 16 מגשים ממוספרים (0, 1 ,...15). בכל מגש יש מקום ל- 32 כדורים בשורה אחת. אלו הם "<u>האוגרים</u>" הרגילים. על השולחן יש עוד כמה אוגרים (של 64 כדורים בשורה אחת) הנקראים "האוגרים למספרים עם נקודה צפה". מול האדם ישנו ארון ענק ובו הרבה מגירות ממוספרות (0, 1, 2,3, ...). בכל מגירה יש מגש שיש בו מקום ל- 8 כדורים בשורה אחת.

ארון זה הוא "<u>הזכרון</u>". כל מגרה בזכרון דומה בהרכבה לתור או הבית (CHARACTER/BYTE) שתואר בסרט נייר מחורר של 8 ערוצים. כך אפשר לראות את הזכרון כסדרת תורים.

לאדם ליד שולחן הבקרה יש שליח המבצע את הוראתו. הנה הוראה ראשונה: לך לארון הזכרון ותתחיל לבצע את ההוראות החל ממגרה 614. השליח ניגש למגרה, פותח אותה ומוצא סדרה של 8 מקומות תפוסים במלואם או בחלקם (או ריקים) בכדורים. השליח שואל מה עליו לעשות. התשובה היא פשוטה יש לקרוא למקום הריק "0"-אפס- ולמקום המלא "1" - אחד. ישנם 256 אפשרויות לסדרה של 8 ספרות של אפסים ואחדים. ישנה חוברת הוראות המסבירה איך לבצע "<u>הוראה</u>" שהיא סדרה כזו. הדבר הראשון המופיע בהסבר ההוראה בחוברת הוא כמה מגרות זכרון נוספות יש לקחת כדי להשלים את ההוראה (אם רק מגירה אחת נוספת, אזי שולפים גם מגירה 615 ; ואם שלש מגירות נוספות, אזי שולפים מגירות 615, 616, 617). הדבר השני המופיע בהסבר ההוראה הוא מה לעשות ואיך לעשות. ההוראה יכולה להיות להעביר מספר שלם או מספר עם נקודה צפה או מספר דצימלי ממקום מסויים בזכרון לאוגר או לחבר אותו מספר למספר הנמצא באוגר, או לאחסן מספר מהאוגר לזכרון .

משמעות המספרים תוסבר למטה. את כתובת הזכרון ואת כתובת האוגר(ים) שולפים מההוראה.

ההוראה יכולה להיות גם עזיבת הסדר הקודם של ההוראות (ללא תנאי או בתנאי מסויים) והליכה(קפיצה) למקום חדש בזכרון (יותר קדימה או יותר אחורה) כדי להמשיך לבצע את ההוראות.

בדוגמא הנ"ל אם ההוראה הראשונה תופסת שתי מגירות (614, 615), בגמר ביצועה נגשים להוראה המתחילה במגירה 616. בדרך יכולים להיתקל בהוראה כזו : אם המספר הנמצא באוגר 3 הוא חיובי, יש לעזוב את סדר ההוראה הנוכחי ולהתחיל לבצע החל ממגרה 312 בזכרון.

אם נחליף את הכדורים במטען אלקטרוני, נבין את יסודות עבודת המחשב.

ב. המשמעויות האפשריות לבית אחד או סדרת בתים בזכרון

1) הוראה : לפי התאור הנ"ל

2) סדרת תווים : למשל 11110010 משמעותה הספרה 2 ;

11000001 משמעותה האות A ;

01000000 משמעותה התו רווח.

3) סדרה לוגית : כל סבית מתוך שמונת הסביות המרכיבות את התו/ הבית תקבל הגדרה לוגית - למשל ב-"0" בסמן חוסר השכלה גבוהה וב-"1" בסמן השכלה גבוהה של העובד.

4) מספר בינרי שלם (2 בתים או ארבעה בתים רצופים) : ערך הספרה "1" מימין לשמאל יהיה כך : 1, 2, 4, 8,...וכו'. לכן המספר הבינרי 1011 שווה ל-11 (= 1+2+8). את המספר השלילי מקבלים ע"י חסור המספר החיובי ממספר אפס (כאשר לווים 2 מהספרה השמאלית השכנה, במקום 10 באריתמטיקה רגילה).

5) מספר עם נקודה צפה (4 בתים או 8 בתים רצופים). מגדירים את המספר ע"י שני נתונים: שבר וחזקה. על בסיס 10 למשל יהיה המספר 64 שווה ל- 0.64×10^2 . כאן השבר הוא 0.64 והחזקה היא 2. אותו מספר על בסיס 16 שווה ל- 0.25×16^2 . כאן השבר הוא 0.25 והחזקה היא 2. ברור שגם החזקה וגם השבר יכולים להיות שליליים.

6) מספר דצימלי : אפשר להכניס ספרה דצימלית בכל תו או לדחוס 2 ספרות דצימליות בתו אחד.

ג. <u>כתיבת תוכניות למחשב</u>

כדי לבצע פעולה מסוימת של חישוב או עבוד נתונים, יש לכתוב סדרת הוראות ולשכן אותו בזכרון. הכתיבה אפשרית בשלש רמות :

1) כתיבה בשפה היסודית של המחשב : דהיינו כל תור הוראה יכתב בשמוגה ספרות בינריות (0 & 1). כדי להקל על הכתיבה, אפשר לכתוב ספרה הקסדצימלית אחת במקום 4 ספרות בינריות כדלקמן :

0000	0001	0010	0011	0100	0101	0110	0111
0	1	2	3	4	5	6	7
1000	1001	1010	1011	1100	1101	1110	1111
8	9	A	B	C	D	E	F

2) כתיבה בשפת האסמבלר : זוהי כתיבה קלה יותר מהנ"ל. יש ביטוי סימבולי לכל הוראה וכן ביטוי סימבולי למקומות התפוסים בזכרון. הכתיבה בשפת האסמבלר מקבילה בערך לכתיבה בשפה היסודית.
ישנה תוכנית במערכת הידועת לתרגם תוכנית באסמבלר לתוכנית בשפה היסודית.

3) כתיבה בשפה עילית : כאן כותבים בשפה דומה לשפה רגילה(אנגלית) ומשתחררים מהליכה במקביל להוראות היסודיות.
ישנה תוכנית הידועת לתרגם תוכנית בשפה העילית לתוכנית בשפת האסמבלר ובסוף לתוכנית בשפת המכונה. מבין השפות העיליות הנמצאות בשמוש רחב נציין : FORTRAN , COBOL , PL1, BASIC

4. תכנות בשפת ה- FORTRAN

א. תרגיל חשוב על הנייר :

מחשוב שורש רבוע של מספר נתון, נלמד את יסודות הכתיבה בשפת הפורטרן. נניח שהמספר הוא 144. נניח שורש נסיוני 1. נחלק 144 ל-1 ונקבל 144 כשורש נסיוני נגדי(שני). נחשב ממוצע שני השרשים $(\frac{144+1}{2})$ דהיינו 72.5. עכשיו ממוצע זה ישמש שורש נסיוני ראשון. נמשיך בפעולה זו כמה פעמים כדלקמן :

צעד	המספר הנתון	שורש נסיוני ראשון	שורש נסיוני שני	ממוצע השרשים
1	144.00	1.00	144.00	72.50
2	144.00	72.50	1.99	37.25
3	144.00	37.25	3.87	20.56
4	144.00	20.56	7.00	13.78
5	144.00	13.78	10.45	12.11
6	144.00	12.11	11.89	12.00
7	144.00	12.00	12.00	12.00

ב. כתיבת התרגיל הנ"ל בשפת הפורטרן :

```
REAL   X, R1, R2, D
X  = 144.00
R1 = 1.0
R2 = X/R1
R1 = (R1+R2)/2.00
```

} לחזור 7 פעמים על שתי השורות

כל שורה תהיה מנוקבת על כרטיס אחד החל מטור 7 (כל ניקוב בטור 6 משמעותו שהכרטיס הנו המשך לכרטיס הקודם. חמשת הטורים הראשונים משמשים למספר התייחסות לשורה). בשורה הראשונה אנו מכריזים על תפיסת מקום בזכרון למספרים עם נקודה צפה(REAL בשפת ה- FORTRAN). מקום אחד ישא את השם X ומיועד להחזיק את המספר הנתון. R1 , R2 מיועדים לשורש הראשון ולשורש השני. D מיועד להפרש ביניהם. לזכור ששם סימבולי של מקום חייב להתחיל באות ויכול להכיל ספרות - אורכו עד 7 תווים. הסימן "=" נקרא "יהיה שווה ל-". הסימן "+" לחבור ; הסימן "-" להסור הסימן "*" לכפל ; הסימן "/" לחלוקה ; והסימנים "**" לחזקה. השמוש בסוגריים דומה לזה באלגברה.

ג. <u>חסכון בכתיבה חוזרת של צעדים – שיטת ספירת הצעדים</u>

להוסיף בהתחלת התוכנית את השורה הבאה שמשמעותה תפיסת מקום בזכרון להחזקת מספר שלם בשם הסימבולי I שישמש לספירת הצעדים :

INTEGER I

ובמקום שתי השורות שיש לחזור עליהן בכתיבה (בניקרב) שבע פעמים בכתוב :

```
      DO  321  I= 1,7
      R2 = X/R1
321   R1 = (R1+R2)/2.00
```

המספר 321 שהוזכר פעמים משמש להתיחסות בלבד.

ד. <u>חסכון בכתיבה חוזרת של צעדים – שיטת התנאי</u>

הפעם לא נחזור על מספר קבוע של צעדים אלא נמשיך לחזור עד שנגיע ההפרש בין שני השרשים קטן מ- 0.001 . הנה התחליף לשתי השורות החוזרות :

```
12    R2 = X/R1
      D  = (R1-R2)*(R1-R2)
      R1 = (R1+R2)/2.00
      IF (D .GT. 0.001) GØ TØ 12
```

בשורה האחרונה "GØ TO 12" משמעותה לחזור לבצע החל מהשורה המסומנת 12. הסימן .GT. משמעותו גדול מ-(GREATER THAN). שאר הסימנים במשפחה זו הם :

NOT EQUAL = .NE. ; EQUAL = .EQ. ; LESS THAN = .LT.

LESS OR EQUAL= .LE. ; GREATER OR EQUAL = .GE.

ה. קלט ופלט

כדי לא לכתוב את התוכנית מחדש עבור כל מספר נתון, אפשר לקרוא את המספר הנתון מכרטיס חיצוני (נניח שמנוקב בטורים 1 עד 8 : 144.00bb). בתוכנית הנ"ל במקום השורה X=144.00 נכתוב :

```
      READ  (5,1)     X
1     FORMAT(F8.2)
```

משמעות השורה הראשונה היא לקרוא מבחוץ. יש לענות כאן על שלש שאלות : איפה לקרוא, איך לקרוא, ומה לקרוא. "5" בפורטרן משמעותו לקרוא מקורא-כרטיסים. המספר "1" בתוך הסוגריים בשורת ה- READ מתייחס לשורת הפורמט המסבירה איך לקרוא - הפרוש שניתן בשורת ה- FORMAT הוא שלפנינו מספר בנקודה צפה (F= FLOATING) שתופס 8 תווים, כאשר ישנן שתי ספרות אחרי הנקודה העשרונית. בסוף שורת ה- READ כתוב מה לקרוא : דהיינו המספר שנקרא יכנס ל- X .

אם רוצים להדפיס את התוצאה - המספר המקורי והשורש שלו - אפשר להוסיף בסוף התוכנית את שתי השורות הבאות :

```
        WRITE (6,2)  X, R1
2       FORMAT(' THE SQUARE ROOT OF',F8.2,'bIS',F7.3)
```

- במקביל לקריאה, אנו אומרים בשורת הכתיבה (WRITE), איפה לכתוב, איך לכתוב ומה לכתוב. "6" בפורטרן משמעותו כתיבה על המדפסת.

ו. תפיסת סדרה של מספרים (שמור בוקטור)

נניח שאנו רוצים להוציא את שורש הריבוע עבור שלשה מספרים. לא כדאי לנו לחזור על כל הכתוב שלש פעמים. במקום להכריז על X נכריז על X(3) שמשמעותו סדרה (וקטור) של שלשה מספרים הנקראים סימבולית בשמות X(1), X(2), X(3). נניח שנקרא את כל שלשת המספרים מכרטיס חיצוני במבנה הקודם, כאשר כל מספר תופס 8 תווים (כולל הרווחים, אם ישנם, משמאל). התוכנית המלאה עם הקלט והפלט תהיה כדלקמן :

```
            REAL    X(3), R1,R2,D
            INTEGER N
1           FORMAT  (3 F8.2)
2           FORMAT  (' THE SQUARE ROOT OF',F8.2, 'bIS',F7.3)
            READ    (5,1) X
            DO   50  N = 1,3
            R1 = 1.0
12          R2 = X(N)/R1
            D  = (R1-R2)*(R1-R2)
            R1 = (R1+R2)/2.0
            IF   (D  .GT.  0.001) GO TP 12
            WRITE(6,2)  X(N), R1
50          CONTINUE
```

השורה האחרונה משמעותה לא לעשות כלום; משמשת רק כדי לשאת מספר התייחסות עבור החזרה.

5. תכנות בשפת ה- PL1

א- תרגיל ב- PL1

את התרגיל האחרון בפרק הקודם ניתן לכתוב בשפת ה- PL1 כדלקמן :

```
PROGRAM : PROC OPTIONS (MAIN) ;
          DECLARE ( X(3),R1,R2,D ) DECIMAL FLOAT (16) ;
          DECLARE N BINARY FIXED (31,0);
IN      : FORMAT (3 F (8,2) ;
OUT     : FORMAT (SKIP,A,F(8,2),A,F(7,3)) ;
          GET EDIT(X) (R(IN) );
          DO N=1 TO 3;
             R1=1.0;
             REPEAT : R2=X(N)/R1;
             D=(R1-R2)*(R1-R2);
             R1=(R1+R2)/2.0;
             IF D> 0.001 THEN GO TO REPEAT;
             PUT EDIT ('THE SQUARE ROOT OF',X(N),' IS',R1) (R(OUT) );
          END;
STOP; END;
```

ב - <u>תאור כללי לשפת ה- PL1</u>:

מהתרגיל הנ'ל נבסה לעמוד על כמה דברים המאפיינים את השפה. ב- PL1 כותבים החל מטור 2 בכרטיס (עד טור 72). כל משפט מסתיים בתור " ;" . אפשר לכתוב יותר ממשפט אחד בכרטיס וכן להמשיך את המשפט בכרטיס הבא .

משפט- הצהרה על משתנים מתחילים במלה DECLARE (או בקיצור DCL) המקביל למספר שלם בפורטרן הוא (31,0) BINARY FIXED שמשמעותו מספר שלם בינרי באורך 31 ספרות כאשר ~~ישנן אפס~~ אין 31 ספרות מימין לנקודה הבינרית. מספר עם נקודה צפה יכל להיות DECIMAL FLOAT או BINARY FLOAT.

הפורמטים ב- PL1 דומים לאלו בפורטרן. F(8,2) ב-PL1 במקום F (8.2) בפורטרן. משמעות ה- SKIP הוא להתחיל להדפיס בשורה חדשה. "A" המופיע ב-FORMAT מובנו שטח של תווים.

בשורה הראשונה בתוכנית PL1 נותנים את שם התוכנית, כדי לזהות משפט מתייחסים אליו בשם (החייב להתחיל באות) ולא במספר דצימלי כמו בפורטרן. אחר שם כזה מופיע התו ":" .

לקריאת/כתיבת רשומות משתמשים ב- PL1 בהוראות WRITE/READ לעומת זאת משתמשים ב- PUT/GET לקריאת/כתיבת סדרת תווים (אשר בידינו לחלקם לשורות).

PUT/GET באות באחת משלש צורות : עם DATA כדי לקרוא/לכתוב משתנים/נתונים בצורת A=4.5 , B='ABC' ; עם LIST ללא שם המשתנה וללא סימן "=" ; ועם EDIT לקריאה/כתיבה בפורמטים.

ב-PUT EDIT / GET EDIT קוראים/כותבים לפי פורמט בצורה קצת שונה מפורטרן : כותבים בהתחלה את כל הדברים שיש לקרוא/לכתוב (כולל סדרות התווים שאנו רוצים לתת, מוזרץ למשתנים) ואחר-כך את הפורמט. אם הפורמט לא מופיע באותו משפט - אלא רחוק ובנפרד, מזכירים אותו עם האות R (המסמלת REMOTE , רחוק).

הצעדים שחוזרים עליהם בשיטת DO מסתיימים ב -"END" .

6. תכנות בשפת ה- COBOL

א- תרגיל ב- COBOL :

את התרגיל האחרון שנכתב בשתי השפות FORTARN , PL1 ניתן לכתיבה בשפת ה-COBOL כדלקמן :

```
.ID  DIVISION
PROGRAM-ID. EXERCISE-COBOL
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION
     01  N  PIC 999.
     01  R1 PIC 99999V99999.
     01  R2 PIC 99999V99999.
     01  D  PIC 99999V99999.
     01  RR1 PIC ZZZZ9.9999.
     01  DD1 PIC ZZZZ9.9999.
     01  VA.
      02  VECD OCCURS 3.
       03 VBA PIC 9(5) .
       03 VBC PIC 9(2) .
     01 VC REDEFINES VA.
      02 VEC  PIC 9(5)V9(2) OCCURS 3.
     01 VB.
      02 VECB OCCURS 3.
       03 VBA PIC 9(5).
       03 VBB PIC X.
       03 VBC PIC 9(2).
     01 CARD.
      02 CA PIC X(8).
      02 CB PIC X(8).
      02 CC PIC X(8).
```

```
PROCEDURE DIVISION.
    ACCEPT CARD.
    MOVE CA TO VECB(1).
    MOVE CB TO VECB(2).
    MOVE CC TO VECB(3).
    PERFORM KELET THRU REPEET VARYING N FROM 1 BY 1 UNTIL N = 4.
       GO TO SOF.
KELET. MOVE CORR VECB (N) TO VECD (N).
HISHUV. COMPUTE R1 =1.
REPEET. COMPUTE R2 = VEC (N) / R1.
        COMPUTE D = (R1 - R2) * (R1 - R2).
        COMPUTE R1 = (R1 + R2) / 2.
        IF D > 0.001 GO TO REPEET
        MOVE R1 TO RR1.
        MOVE VEC (N) TO DD1.
        DISPLAY 'THE SQURE ROOT OF',DD1, 'IS',RR1.
SOF.    STOP RUN.
```

ב - <u>תאור כללי לשפת ה- COBOL</u> :

מהתרגיל הנ׳ל ננסה לעמוד על הדברים המאפיינים את השפה.
תוכנית בשפת ה-COBOL נכתבת בצורה דומה לחבור באנגלית. התוכנית
מחולקת לארבע חטיבות (DIVISIONS):

IDENTITY DIVISION לזיהוי התוכנית

ENVIRONMENT DIVISION לתאור מערכת קלט-פלט

DATA DIVISION לתאור הנתונים

PROCEDURE DIVISION לבצוע התוכנית.

החטיבה יכולה להתחלק למחלקות (SECTIONS). למשל חטיבת הנתונים יכולה להכיל את מחלקת הקבצים (FILE SECTION) ואת מחלקת שטח-העבודה (WORKING STORAGE SECTION).

המחלקות יכולות להחלק לפסקות (PARAGRAPHS) כאשר הפסקה מתחילה בשם (LABEL) ומסתיימת לפני השם הבא : הפסקות בנויות ממשפטים (SENTENCES). כל משפט מסתיים בנקודה. בד"כ המשפט עשוי מתורי (STATEMENT) ובנוי ממלים (WORDS).

כרטיסי ה-DIVISION , SECTION , LABEL מתחילים מטור 8 (זה נקרא A-MARGIN); וכרטיסי הפקודות והמשתנים מתחילים מטור 12 (B-MARGIN).

כרטיס המנוקב מטור 7 משמעותו כרטיס הערה.

הנקודה העשרונית במספר המנוקב על כרטיס מהווה בעיה בקריאתו. לכן קוראים בהתחלה את כל המספר כסדרת תווים, ואחר כך מעבירים את המספר ללא-נקודה עשרונית למשתנה של מספרים.

COBOL איננה נוחה לחישובים, אך היא די טובה לעיבוד נתונים (בטפול ברשומות ובשדות בתוך הרשומות). ביסודה היא שפה מסחרית.

השפה מדקדקת מאוד בסימני הניקוד והרווחים. רווח חייב להופיע לפני ואחרי הסימנים (> < / * - + =).

השם הראשי של משתנה מופיע עם קדומת 01 לפניו. את הוא מחולק למשתנים משניים תופיע הקדומת 02 לפני כל שם. המשתנה המשני יכל להתחלק גם הוא למשתנים תת-משניים, שיופיעו עם קדומת 03 וכו'.

בתרגיל הנ"ל מופיעה המלה PIC במקום PICTURE המתארת את המבנה של המשתנה : 9 במקום תו שהוא ספרה ; V עבור נקודה עשרונית דמיונית; Z עבור 0 המופיע כרווח ; . עבור נקודה עשרונית להדפסה ; X עבור תו כלשהו.

המלה OCCURS משמשת לציון וקטור.

המלה REDEFINES מאפשרת לתת שם נוסף למשתנה וכן חלוקה חדשה לתת-משתנים.

MOVE CORR (קצור של MOVE CORRESRONDING) להעביר תוכן משתנה למשתנה אחר לפי שמות הזהוי של התת-משתנים.

DISPLAY היא הפקודה להדפסה.

7- תכנות בשפת ה-BASIC

א-תרגיל ב-BASIC

את התרגיל האחרון שנכתב בשלש השפות FØRTRAN, PL1, CØBØL, ניתן לכתוב בשפת ה-BASIC כדלקמן :

```
EXERCISE
010  REM   THIS IS AN EXERCISE IN BASIC
020  READ  X(1), X(2), X(3)
030  FØR   N=1  TØ 3
040  LET   R1 = 1.0
050  LET   R2 = X(N)/R1
060  LET   D=(R1+R2)*(R1-R2)
070  LET   R1=(R1+R2)/2.0
080  IF  D > 0.001  GØ  TØ  050
090  PRINT " THE SQUARE RØØT ØF", X(N), "IS", R1
100  NEXT  N
110  DATA  0.286989, 17.9311, 297.783
120  END
```

בצורה הנ"ל קוראים שלשת המספרים, למטרת חישוב השורש הרבועי, מכרטיס נתונים - שהוא כרטיס המתחיל במספר המזהה 110 שאחריו המלה DATA.
אם רוצים להפעיל את התוכנית ממסוף ולספק את הנתונים בעת ההפעלה, אזי אין צורך בכרטיס ה-DATA. לעומת זאת יש להחליף כרטיס (020) לצורה הבאה :

```
020  INPUT  X(1), X(2), X(3)
```

כדי להפעיל את התוכנית במסוף לאחר יצירתה, יש להדפיס בהתחלת השורה את ההוראה RUN

ואז תופיע שורת הדפסה המכילה את שם התוכנית EXERCISE
אחריה יופיע סימן שאלה בהתחלת השורה ?

אחרי סימן זה, יש לספק את הנתונים כדלקמן :

```
? 0.286989, 17.9311, 297.783
```

ומסיימים בלחיצה על המקש RETURN.
התוצאה תופיע במסוף כדלקמן :

```
THE SQUARE RØØT ØF 0.286989  IS 0.535714
THE SQUARE RØØT ØF 17.9311   IS 4.23451
THE SQUARE RØØT ØF 297.783   IS 17.2564
DØNE
```

ב- תאור כללי לשפת ה-BASIC

השפה דומה בחלקה ל-FØRTRAN ובחלקה ל- PL1. היא יותר פשוטה משתיהן (וגם פחות עשירה).
כל כרטיס מתחיל במספר עשרוני מזהה.
הכרטיס האחרון הוא כרטיס END אשר מספרו חייב להיות גבוה מכל שאר המספרים.
כרטיס הערה מתחיל במלה REM (אחרי המספר המזהה).
כל משפט הצבה מתחיל במלה LET (במקום X=4 ב- FØRTRAN וב-PL1, כותבים כאן LET X=4).
בסימני השוויון/אי-השוויון משתמשים ב- =, <>, <, >, <=, >=
מדפיסים בהוראת PRINT וקוראים בהוראה READ מכרטיסי DATA, או בהוראה INPUT ממסוף.
הלולאה מתחילה במשפט FØR וגומרת במשפט NEXT. למשל :

```
FØR  I=1  TØ 11  STEP 2
......
......
NEXT  I
```

כאן הספירה בלולאה נעשית ע"י I אשר מתקדם בצעדים של 2
(אם הקדום הוא ב- 1, אין צורך לכתוב STEP).